



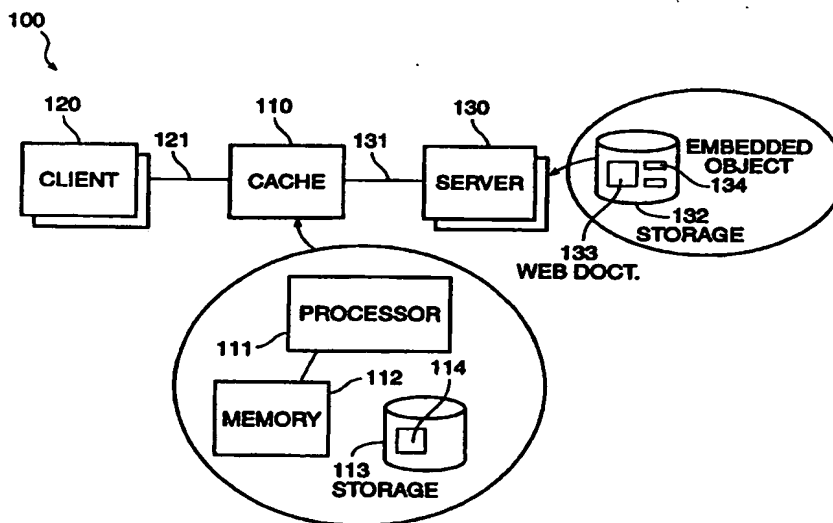
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>G06F 17/30</b>	<b>A1</b>	(11) International Publication Number: <b>WO 99/22316</b> (43) International Publication Date: <b>6 May 1999 (06.05.99)</b>
(21) International Application Number: <b>PCT/US98/21008</b> (22) International Filing Date: <b>2 October 1998 (02.10.98)</b> (30) Priority Data: 08/959,313                      28 October 1997 (28.10.97)                      US (71) Applicant: <b>CACHEFLOW, INC. [US/US]; 650 Almanor Avenue, Sunnyvale, CA 94086 (US).</b> (72) Inventors: <b>CROW, Doug; 24, 133 S.E. 45th Place, Issaquah, WA 98029 (US). BONKOWSKI, Bert; 345 Coleridge Drive, Waterloo, Ontario N2L 3E6 (CA). CZEGLIEDI, Harold; 199 Ironwood Place, Waterloo, Ontario N2T 2L4 (CA). JENKS, Tim; 5636 42nd Avenue, S.W., Seattle, WA 98136 (US).</b> (74) Agent: <b>SWERNOFSKY, Steven, A.; The Law Offices of Steven A. Swernofsky, P.O. Box 390013, Mountain View, CA 94039-0013 (US).</b>	(81) Designated States: <b>CA, CN, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</b>  Published <i>With international search report.</i>	

(54) Title: **SHARED CACHE PARSING AND PRE-FETCH**

## (57) Abstract

The invention provides a method and system for reducing latency in reviewing and presenting web documents to the user. A cache coupled to one or more web clients request web documents from web servers on behalf of those web clients and communicates those web documents to the web clients for display. The cache parses the web documents as they are received from the web server, identifies references to any embedded objects, and determines if those embedded objects are already maintained in the cache. If those embedded objects are not in the cache, the cache automatically pre-fetches those embedded objects from the web server without need for a command from the web client. The cache maintains a two-level memory including primary memory and secondary mass storage. At the time the web document is received, the cache determines if any embedded objects are maintained in the cache but are not in primary memory. If those embedded objects are not in primary memory, the cache automatically pre-loads those embedded objects from secondary mass storage to primary memory without need for a request from the web client. Web documents maintained in the cache are periodically refreshed, so as to assure those web documents are not stale. The invention is applied both to original requests to communicate web documents and their embedded objects from the web server to the web client, and to refresh requests to communicate web documents and their embedded objects from the web server to the cache.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## Shared Cache Parsing and Pre-fetch

Background of the Invention1. *Field of the Invention*

This invention relates to caches.

2. *Related Art*

When presenting and reviewing data using a web browser or web client, that is, a client program for the web (the "World Wide Web") such as Netscape Corporation's "Navigator" product or Microsoft Corporation's Internet Explorer" product, it is desirable to present the data with as little delay as possible. If the user of the web client has to wait too long for the data to be displayed, this can lead to user dissatisfaction.

Some web clients access the web using a proxy cache, that is, a device for requesting web documents on behalf of the web client and for caching those web documents for possible later use. The proxy cache acts to reduce the amount of communication bandwidth used between the web client and web servers. A proxy cache can be shared by more than one web client, in which case it acts to reduce the total amount of communication bandwidth used between all of its web clients and web servers. One advantage of the proxy cache is that web documents stored in cache can be accessed more quickly than re-requesting those web documents from their originating web server.

One problem in the art is that a document requested by the web client (a "web document") can include, in addition to text and directions for display, embedded objects which are to be displayed with the web document. Embedded objects can include pictures, such as data in GIF or JPEG format, other multimedia data, such as animation, audio (such as streaming audio), movies, video (such as streaming video), program fragments, such as Java, Javascript, or ActiveX, or other web documents, such as when using frames. The web client must parse the web document to determine the embedded objects, and then request the embedded objects from the web server.

1 While using a proxy cache ameliorates this problem somewhat, the problem per-  
2 sists. If there are many embedded objects in the web document, it can take substantial time to  
3 identify, request, communicate, and display all of them. Parsing and requesting embedded ob-  
4 jects by the web client is serial, and most web clients are set to request only a small number of  
5 embedded objects at a time. Web clients requesting embedded objects perform this task in par-  
6 allel with rendering those objects for display, further slowing operation.

7  
8 Moreover, known proxy caches use a two-level memory having a both primary  
9 memory and secondary mass storage. Even those embedded objects already maintained in the  
10 cache, and thus accessible by the web client without requesting them from the web server, can  
11 have been dropped out of the primary memory to secondary mass storage, possibly delaying  
12 communication of the embedded objects from the proxy cache to the web client and thus de-  
13 laying display of those embedded objects to the user.

14  
15 Accordingly, it would be advantageous to provide a method and system for re-  
16 ducing latency in reviewing and presenting web documents to the user. This advantage is  
17 achieved in a system in which web documents are parsed by a cache for references to embedded  
18 objects, and those embedded objects are pre-fetched from the web server or pre-loaded from  
19 secondary mass storage by the cache before they are requested by the web client.

20  
21 Teachings of the art include (1) the known principle of computer science that de-  
22 vices work better when they are indifferent to the nature of the data they process, and (2) the  
23 known principle of client-server systems that it is advantageous to assign processing-intensive  
24 tasks to clients, rather than to servers, whenever possible. The invention is counter to the first  
25 teaching, as the cache alters its behavior in response to its parsing of the web documents it re-  
26 ceives for communication to the client. The invention is also counter to the second teaching, as  
27 the cache takes on the additional processing tasks of parsing the web document for embedded  
28 objects and, if necessary, independently requesting those embedded objects from the web  
29 server.

### 30 31 Summary of the Invention

32  
33 The invention provides a method and system for reducing latency in reviewing  
34 and presenting web documents to the user. A cache coupled to one or more web clients request  
35 web documents from web servers on behalf of those web clients and communicates those web

1 documents to the web clients for display. The cache parses the web documents as they are re-  
2 ceived from the web server, identifies references to any embedded objects, and determines if  
3 those embedded objects are already maintained in the cache. If those embedded objects are not  
4 in the cache, the cache automatically pre-fetches those embedded objects from the web server  
5 without need for a command from the web client.  
6

7 In a preferred embodiment, the cache maintains a two-level memory including  
8 primary memory and secondary mass storage. At the time the web document is received, the  
9 cache determines if any embedded objects are maintained in the cache but are not in primary  
10 memory. If those embedded objects are not in primary memory, the cache automatically pre-  
11 loads those embedded objects from secondary mass storage to primary memory without need  
12 for a request from the web client.  
13

14 In a preferred embodiment, web documents maintained in the cache are periodi-  
15 cally refreshed, so as to assure those web documents are not "stale" (changed at the web server  
16 but not at the cache). The invention is applied both to original requests to communicate web  
17 documents and their embedded objects from the web server to the web client, and to refresh re-  
18 quests to communicate web documents and their embedded objects from the web server to the  
19 cache.  
20

### 21 Brief Description of the Drawings

22

23 Figure 1 shows a block diagram of a system for shared cache parsing and pre-  
24 fetch.  
25

26 Figure 2 shows a flow diagram of a method for shared cache parsing and pre-  
27 fetch.  
28

### 29 Detailed Description of the Preferred Embodiment

30

31 In the following description, a preferred embodiment of the invention is described  
32 with regard to preferred process steps and data structures. Those skilled in the art would recog-  
33 nize after perusal of this application that embodiments of the invention can be implemented  
34 using one or more general purpose processors or special purpose processors or other circuits  
35 adapted to particular process steps and data structures described herein, and that implementa-

tion of the process steps and data structures described herein would not require undue experimentation or further invention.

Inventions disclosed herein can be used in conjunction with inventions disclosed in one or more of the following patent applications:

Provisional U.S. Application 60/048,986, filed June 9, 1997, in the name of inventors Michael Malcolm and Robert Zarnke, titled "Network Object Cache Engine," assigned to CacheFlow, Inc., attorney docket number CASH-001.

U.S. Application Serial No. 08/\_\_\_\_\_, filed this same day, in the name of inventors Michael Malcolm and Ian Telford, titled "Adaptive Active Cache Refresh," assigned to CacheFlow, Inc., attorney docket number CASH-003.

These applications are referred to herein as the "Cache Disclosures," and are hereby incorporated by reference as if fully set forth herein.

### *System Elements*

Figure 1 shows a block diagram of a system for shared cache parsing and pre-fetch.

A system 100 includes a cache 110, at least one client device 120, and at least one server device 130. Each client device 120 is coupled to the cache 110 using a client communication path 121, such as a dial-up connection, a LAN (local area network), a WAN (wide area network), or some combination thereof. Similarly, each server device 130 is also coupled to the cache 110 using a server communication path 131, such as a dial-up connection, a LAN (local area network), a WAN (wide area network), or some combination thereof. In a preferred embodiment, the client communication path 121 includes a LAN, while the server communication path 131 includes a network of networks such as an internet or intranet.

As used herein, the terms "client" and "server" refer to a relationship between the client or server and the cache 110, not necessarily to particular physical devices. As used herein, one "client device" 120 or one "server device" 130 can comprise any of the following:

(a) a single physical device capable of executing software which bears a client or server relation-

1 ship to the cache 110; (b) a portion of a physical device, such as a software process or set of  
2 software processes capable of executing on one hardware device, which portion of the physical  
3 device bears a client or server relationship to the cache 110. The phrases "client device" 120 and  
4 "server device" 130 refer to such logical entities and not necessarily to particular individual  
5 physical devices.

6  
7 The server device 130 includes memory or storage 132 having a web document  
8 133, the web document 133 including references to at least one embedded object 134. In a pre-  
9 ferred embodiment, the web document 133 can include text and directions for display. The em-  
10 bedded object 134 can include pictures such as data in GIF or JPEG format, other multimedia  
11 data, such as animation, audio (such as streaming audio), movies, video (such as streaming  
12 video), program fragments, such as Java, Javascript, or ActiveX, or other web documents, such  
13 as when using frames.

14  
15 The cache 110 includes a processor 111, program and data memory 112, and mass  
16 storage 113. The cache 110 maintains a first set of web objects 114 in the memory 112 and a  
17 second set of web objects 114 in the storage 113. (Web objects 114 can comprise web docu-  
18 ments 13 or embedded objects 134 or both.)

19  
20 In a preferred embodiment, the cache 110 includes a cache device such as de-  
21 scribed in the Cache Disclosures defined herein, hereby incorporated by reference as if fully set  
22 forth therein.

23  
24 The cache 110 receives requests from the client device 120 for a web object 114  
25 and determines if that web object 114 is present at the cache 110, either in the memory 112 or in  
26 the storage 113. If the web object 114 is present in the memory 112, the cache 110 transmits the  
27 web object 114 to the client device 120 using the client communication path 121. If the web  
28 object 114 is present in the storage 113 but not in the memory 112, the cache 110 loads the web  
29 object 114 into the memory 112 from the storage 113, and proceeds as in the case when the web  
30 object 114 was originally present in the memory 112. If the web object 114 is not present in ei-  
31 ther the memory 112 or the storage 113, the cache 110 retrieves the web object 114 from the ap-  
32 propriate server device 130, places the web object 114 in the memory 112 and the storage 113,  
33 and proceeds as in the case when the web object 114 was originally present in the memory 112.

1 Due to the principle of locality of reference, it is expected that the cache 110 will  
2 achieve a substantial "hit rate," in which many requests from the client device 120 for web ob-  
3 jects 114 will be for those web objects 114 already maintained by the cache 110, reducing the  
4 need for requests to the server device 130 using the server communication path 131.

5  
6 The cache 110 parses each web object 114 as it is received from the server device  
7 130, separately and in parallel to any web client program operating at the client device 120. If  
8 the web object 114 is a web document 133 that includes at least one reference to embedded ob-  
9 jects 134, the cache 110 identifies those references and those embedded objects 134, and deter-  
10 mines if those embedded objects 134 are already maintained in the cache 110, either in the  
11 memory 112 or the storage 113.

12  
13 If those embedded objects 134 are not in the cache 110 at all, the cache 110n  
14 automatically, without need for a command from the web client, requests those embedded ob-  
15 jects 134 from the server device 130.

16  
17 The cache 110 has a relatively numerous set of connections to the server commu-  
18 nication path 131, and so is able to request a relatively numerous set of embedded objects 134 in  
19 parallel from the server device 130. Moreover, the cache 110 parses the web document 133 and  
20 requests embedded objects 134 in parallel with the web client at the client device 120 also pars-  
21 ing the web document 133 and requesting embedded objects 134. The embedded objects 134 are  
22 available to the cache 110, and thus to the client device 120, much more quickly.

23  
24 If those embedded objects 134 are maintained in the cache 110, but they are in the  
25 storage 113 and not in the memory 112, the cache 110 automatically, without need for a com-  
26 mand from the web client, loads those embedded objects 134 from the storage 113 into the  
27 memory 112.

28  
29 In a preferred embodiment, those web objects 114 maintained in the cache 110  
30 are periodically refreshed, so as to assure those web objects 114 are not "stale" (changed at the  
31 server device 130 but not at the cache 110). To refresh web objects 114, the cache 110 selects  
32 one web object 114 for refresh and transmits a request to the server device 130 for that web ob-  
33 ject 114. The server device 130 can respond with a copy of the web object 114, or can respond  
34 with a message that the web object 114 has not changed since the most recent copy of the web  
35 object 114 was placed in the cache 110. If the web object 114 has in fact changed, the cache 110



1 proceeds as in the case when a client device 120 requested a new web object 114 not maintained  
2 in the cache 110 at all. If the web object 114 has in fact not changed, the cache 110 updates its  
3 information on the relative freshness of the web object 114, as further described in the Cache  
4 Disclosures.

5  
6 *Method of Operation*

7  
8 Figure 2 shows a flow diagram of a method for shared cache parsing and pre-  
9 fetch.

10  
11 A method 200 includes a set of flow points to be noted, and steps to be executed,  
12 cooperatively by the system 100, including the cache 110, the client device 120, and the server  
13 device 130.

14  
15 At flow point 210, the client device 120 is ready to request a web document 133  
16 from the server device 130. For example, the web document 133 can comprise an HTML page  
17 having a set of embedded objects 134.

18  
19 At a step 221, the client device 120 transmits a request for the web document 133,  
20 using the client communication path 121, to the cache 110.

21  
22 At a step 222, the cache 110 determines if that web document 133 is located in  
23 the memory 112 at the cache 110. If so, the cache 110 proceeds with the step 225. Otherwise,  
24 the cache 110 proceeds with the step 223.

25  
26 At a step 223, the cache 110 determines if that web document 13 is located in the  
27 storage 113 at the cache 110 (but not in the memory 112). If so, the cache 110 loads the web  
28 document 133 from the storage 113 into the memory 112, and proceeds with the step 225. Oth-  
29 erwise, the cache 110 proceeds with the step 224.

30  
31 At a step 224, the cache 110 transmits a request to the server device 130 for the  
32 web document 133. The server device 130 receives the request and transmits the web document  
33 133 to the cache 110. The cache 110 stores the web document 133 in the memory 112 and the  
34 storage 113 and proceeds with the step 225.

1           At a step 225, the cache 110 transmits the web document 133 to the client device  
2   120 for display. In parallel, the cache 110 parses the web document 133 and determines if there  
3   are any references to embedded objects 134. If not, the cache 110 proceeds with the flow point  
4   230. Otherwise, the cache proceeds with the step 226.

5  
6           At a step 226, the cache 110 identifies the embedded documents 134 and repeats  
7   the steps 222 through 226 inclusive (including repeating this step 226) for each such embedded  
8   document 134. Web documents 133 in "frame" format can refer to embedded documents 134  
9   that are themselves web documents 133 and themselves refer to embedded documents 134, and  
10   so on. There is no prospect of an infinite loop if web document 133 is self-referential because  
11   the cache 110 will simply discover at the second reference that the web document 133 is already  
12   maintained in the cache 110.

13  
14           At a flow point 230, the web document 133 and all its embedded objects 134  
15   have been transmitted to the client device 120 for display.

16  
17           When the cache 110 refreshes a web object 114, the cache 110 performs the steps  
18   222 through 226 inclusive (including repeating the step 226) for the web object 114 and for each  
19   identified embedded object 134 associated with the web object 114.

20  
21   *Alternative Embodiments*

22  
23           Although preferred embodiments are disclosed herein, many variations are possi-  
24   ble which remain within the concept, scope, and spirit of the invention, and these variations  
25   would become clear to those skilled in the art after perusal of this application.

Claims

1  
2  
3 1. A method, including the steps of  
4 receiving web documents at a shared cache from a web server or mass storage for  
5 communicating said web documents to a web client for display;  
6 parsing said web documents for references to embedded objects;  
7 determining if said embedded objects are already maintained in said shared  
8 cache;  
9 conditionally pre-fetching said embedded objects from said web server in re-  
10 sponse to said step of determining, without need for a command from said web client.

11  
12 2. A method as in claim 1, including the steps of  
13 maintaining at said shared cache a two-level memory including primary memory  
14 and secondary mass storage;  
15 locating said embedded objects in said shared cache but not in said primary  
16 memory;  
17 conditionally pre-loading said embedded objects from said secondary mass stor-  
18 age into said primary memory in response to said step of locating, without need for a request  
19 from said web client.

20  
21 3. A method as in claim 1, wherein said web documents include refresh  
22 copies of said web documents requested by said shared cache from said  
23 web server.

24  
25 4. A system, including  
26 a shared cache coupled to at least one web server and coupled to a plurality of  
27 web clients, said shared cache being capable of receiving requests for web documents from said  
28 web clients, requesting said web documents from said web server or mass storage, receiving said  
29 web documents from said web server or mass storage, and communicating said web documents  
30 to said web clients;  
31 said shared cache including  
32 means for parsing said web documents for references to embedded objects;  
33 means for determining if said embedded objects are already maintained in said  
34 shared cache; and

means for conditionally pre-fetching said embedded objects from said web server in response to said means for determining, without need for a command from said web client.

5. A system as in claim 4, including

A two-level memory at said shared cache, said two-level memory including primary memory and secondary mass storage;

means for locating said embedded objects in said shared cache but not in said primary memory;

means for conditionally pre-loading said embedded objects from said secondary mass storage into said primary memory in response to said means for locating, without need for a request from said web client.

6. A system as in claim 4, wherein said web documents include refresh copies of said web documents requested by said shared cache from said web server.

7. A shared cache, including

means for parsing said web documents, said web documents being received from a web server or from mass storage, for references to embedded objects;

means for determining if said embedded objects are already maintained in said shared cache; and

means for conditionally pre-fetching said embedded objects from said web server in response to said means for determining, without need for a command from said web client.

8. A cache as in claim 7, including

A two-level memory at said shared cache, said two-level memory including primary memory and secondary mass storage;

means for locating said embedded objects in said shared cache but not in said primary memory;

means for conditionally pre-loading said embedded objects from said secondary mass storage into said primary memory in response to said means for locating, without need for a request from said web client.

9. A cache as in claim 7, wherein said web documents include refresh copies of said web documents requested by said shared cache from said web server.

1/2

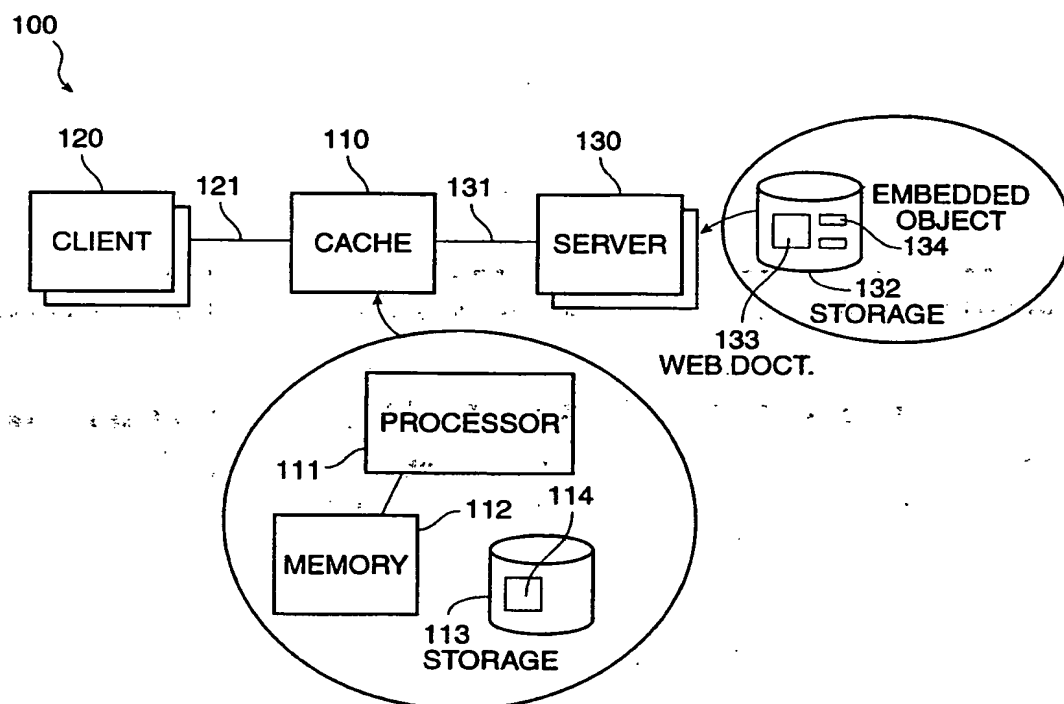


FIG. 1

2/2

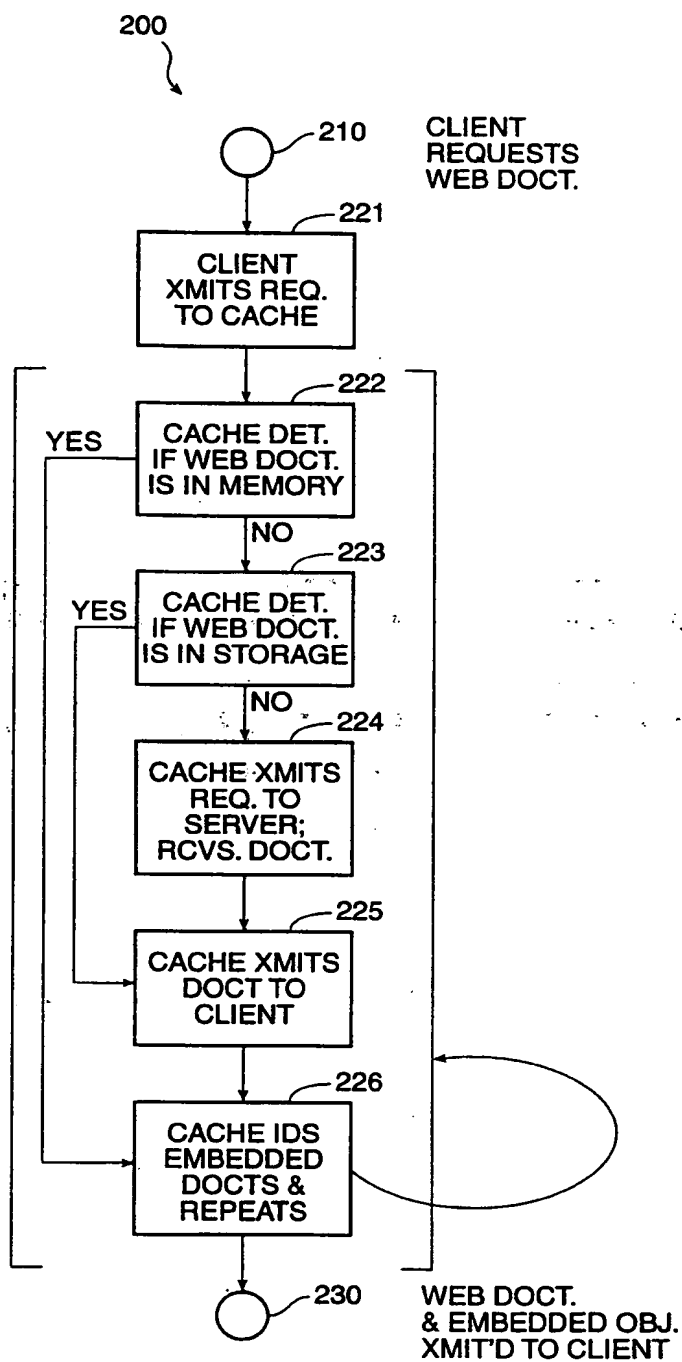


FIG. 2

# INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 98/21008

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>DIAS G. ET AL: "A Smart Internet Caching System"</p> <p>PROCEEDINGS OF THE INET'96 CONFERENCE, MONTREAL, CANADA, 24 - 28 June 1996, XP002086721</p> <p><a href="http://www.isoc.org/inet96/proceedings/a4/a4_3.htm">http://www.isoc.org/inet96/proceedings/a4/a4_3.htm</a></p> <p>see the whole document</p> <p style="text-align: center;">-/-</p>	<p>1,3,4,6,7,9</p>

☒ Further documents are listed in the continuation of box C.

☐ Patent family members are listed in annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"G" document member of the same patent family

Date of the actual completion of the international search

4 December 1998

Date of mailing of the international search report

21/12/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Fournier, C

## INTERNATIONAL SEARCH REPORT

Interr. Application No

PCT/US 98/21008

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>WANG Z ET AL: "Prefetching in World Wide Web" IEEE GLOBECOM 1996. COMMUNICATIONS: THE KEY TO GLOBAL PROSPERITY. GLOBAL INTERNET'96. CONFERENCE RECORD (CAT. NO.96CH35942), IEEE GLOBECOM 1996. COMMUNICATIONS: THE KEY TO GLOBAL PROSPERITY. GLOBAL INTERNET'96. CONFERENCE RECORD, LONDON, UK, 18-22 NO, pages 28-32, XP002086567 ISBN 0-7803-3336-5, 1996, New York, NY, USA, IEEE, USA see page 30, left-hand column, line 44 - page 31, right-hand column, paragraph 7</p>	1,3,4,6,7,9
A	<p>CHINEN K. ET AL: "An interactive Prefetching Proxy Server for Improvement of WWW Latency" PROCEEDINGS OF THE INET'97 CONFERENCE, KUALA LUMPUR, MALAYSIA, 24 - 27 June 1997, XP002086569 <a href="http://www.isoc.org/INET97/proceedings/al/al_3.htm">http://www.isoc.org/INET97/proceedings/al/al_3.htm</a> see the whole document</p>	1,3,7